

Project: impl sm2 with RFC6979

代码说明: 由于 sm2 需要用到 sm3, 先将 sm3 代码复制到本代码中,
根据算法描述编写代码:

算法:

- 1) 产生随机数 k , k 的值从 1 到 $n-1$;
- 2) 计算椭圆曲线点 $C1=[k]G=(x1,y1)$, 将 $C1$ 转换成比特串;
- 3) 验证公钥 PB , 计算 $S=[h] PB$, 如果 S 是无穷远点, 出错退出;
- 4) 计算 $(x2,y2)=[k] PB$
- 5) 计算 $t=KDF(x2||y2, klen)$, KDF 是密钥派生函数, 如果 t 是全 0 比特串, 返回第 1) 步。
- 6) 计算 $C2=M\oplus t$
- 7) 计算 $C3=Hash(x2||M||y2)$
- 8) 输出密文 $C=C1||C3||C2$, $C1$ 和 $C3$ 的长度是固定的, $C1$ 是 64 字节, $C3$ 是 32 字节, 很方便 C 从中提取 $C1$, $C3$ 和 $C2$ 。

版权声明: 本文为 CSDN 博主「独孤木」的原创文章, 遵循 CC 4.0 BY-SA 版权协议, 转载请附上原文出处链接及本声明。

原文链接: <https://blog.csdn.net/boliwu/article/details/81510305>

运行指导：直接运行 py 文件

```
message='helloedu'  
print('要加密的明文为：')  
print(message)  
m1,m2,m3=encrypt(message)  
m=(m1+m2+m3).upper()  
print('加密后的密文为：')
```

加密信息为 message，可手动更改

运行结果截图：

```
要加密的明文为：  
helloedu  
加密后的密文为：  
0467CEA8 C81AC151 D34EA338 4E346CEB EAF9629 C6A074DA B0C8C736 8AE70FCD 296C49AA 28B928EC 1A11A9E9 559118EA D4E8B68C 3C1E202A C1  
BDC0CD 403B3894 3B84441A 525E5DF1 DA0D29F3 FFB5F5B9 C5BADEA2 AE4025EA 12BD6E47 4AE19C74 0310E52A 8A11878A F4  
  
解密后的明文为：  
helloedu
```